

17/3,K/40 (Item 7 from file: 349)
DIALOG(R) File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00551255 **Image available**

A METHOD AND APPARATUS FOR BRANCH PREDICTION USING A SECOND LEVEL BRANCH PREDICTION TABLE

PROCEDE ET DISPOSITIF PERMETTANT LES PREVISIONS DE BRANCHE AU MOYEN D'UNE TABLE DE PREVISION DE BRANCHE DE SECOND NIVEAU

Patent Applicant/Assignee:

INTEL CORPORATION,

YEH Tse-Yu,

SHARANGPANI Harshvardhan P,

Inventor(s):

YEH Tse-Yu,

SHARANGPANI Harshvardhan P,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200014628 A1 20000316 (WO 0014628)

Application: WO 99US19892 19990826 (PCT/WO US9919892)

Priority Application: US 98149885 19980908

Designated States: AE AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE

ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT

LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT

UA UG US UZ VN YU ZA ZW GH GM KE LS MW SD SL SZ UG ZW AM AZ BY KG KZ MD

RU TJ TM AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE BF BJ CF

CG CI CM GA GN GW ML MR NE SN TD TG

Publication Language: English

Fulltext Word Count: 7483

Fulltext Availability:

Claims

Claim

... predictor of claim 1 , further comprising a first circuit coupled to the first BIPT to **predict** an IP address as being a target address stored in a target address cache if...

...the second branch

prediction entry if the first branch prediction entry misses in the first **level** BPT.

16 The branch predictor of claim 15, further comprising the steps of implementing a first type of prediction algorithm in the **first level** BPT and implementing a second type of prediction algorithm in the **second level**

BPT that is different from the first type of prediction algorithm.

17 The method of...

...be a

branch instruction.

. The method of claim 15, wherein the steps of searching the **first level**

BPT and searching the **second level** BPT occur simultaneously.

19 The method of claim 15, wherein the step of searching the **first level** BIPT

includes the step of comparing an address tag of the IP address to an address tag stored in the **first level** BPT, and the step of searching the

second level BPT includes the step of selecting an entry from a directmapped table.

20 A method...

...that the branch is taken.

. The method of claim 20, further comprising the step of **predicting** the

subsequent IP address to be the initial IP address incremented by a predetermined amount...

17/3,K/41 (Item 8 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00494780 **Image available**

PROCESSOR CONFIGURED TO GENERATE LOOKAHEAD RESULTS FROM COLLAPSED MOVES,
COMPARES AND SIMPLE ARITHMETIC INSTRUCTIONS
PROCESSEUR CONFIGURE POUR GENERER DES RESULTATS D'ANTICIPATION ET POUR
REDUIRE LES DEPLACEMENTS ET LES COMPARAISONS, ET INSTRUCTIONS
ARYTHMETIQUES SIMPLES RECEVANT LES RESULTATS D'ANTICIPATION

Patent Applicant/Assignee:

ADVANCED MICRO DEVICES INC,

Inventor(s):

WITT David B,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9926132 A2 19990527

Application: WO 98US22030 19981019 (PCT/WO US9822030)

Priority Application: US 9765878 19971117; US 98115123 19980714

Designated States: JP KR AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT
SE

Publication Language: English

Fulltext Word Count: 13232

Fulltext Availability:

Detailed Description

Detailed Description

... includes 40 lines, each of which may store up to six execution results corresponding to **concurrently** dispatched instructions. Execution results are retired from the result queue in order into the register...
TLB similar to LI I-cache 14.

External interface unit 42 is configured to transfer **cache** lines of instruction bytes and data bytes into processor 10 in response to **cache** misses. Instruction **cache** lines are routed to predecode unit 12, and data **cache** lines are routed to LI D- **cache** 3 8. Additionally, external interface unit 42 is configured to transfer **cache** lines 3 5 discarded by L 1 D- **cache** 3 8 to memory if the discarded **cache** lines have been modified to processor 1 0. As shown in Fig. 1, external interface unit 42 is configured to interface to an external L2 **cache** via L2 interface 44 as well as to interface to a computer system via bus...

...history table 60, a branch select mux 62, a return stack 64, an indirect address **cache** 66, and a forward collapse unit 68. Fetch control unit 50 is coupled to LI I- **cache** 14, LO I- **cache** 16, indirect address **cache** 66, return stack 64, branch history table 60, branch scanner 58, and instruction select mux...prediction information (including target addresses and taken/not taken predictions) from branch scanner 58, branch **history** table 60, return stack 64, and indirect address cache 66.

Responsive to the branch prediction...

17/3,K/42 (Item 9 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00482052 **Image available**

DATA PROCESSING SYSTEM WITH MULTI-WAY SET ASSOCIATIVE CACHE

SYSTEME INFORMATIQUE A ANTEMEMOIRE ASSOCIATIVE PAR TABLES A PLUSIEURS VOIES

Patent Applicant/Assignee:

INTERNATIONAL BUSINESS MACHINES CORPORATION,

Inventor(s):

LEVENSTEIN Sheldon Bernard,

KROLAK David John,
Patent and Priority Information (Country, Number, Date):
Patent: WO 9913404 A1 19990318
Application: WO 98US17576 19980825 (PCT/WO US9817576)
Priority Application: US 97924272 19970905
Designated States: CA CN IL IS JP KR AT BE CH CY DE DK ES FI FR GB GR IE IT
LU MC NL PT SE
Publication Language: English
Fulltext Word Count: 14349

Fulltext Availability:
Detailed Description

Detailed Description

... in the set references that memory address. Typically, this is performed by either sequentially or **concurrently** comparing the tag for a given memory address with the tag for each directory entry...a memory access request from virtual addressing to real addressing is also required for many **cache** accesses.

It may be possible to utilize class prediction algorithms to attempt to predict a...

...history array accessed by virtual address bits to control the late select of an internal **level one cache** without penalizing the cycle time of the **cache**.

However, these designs are not well suited for external **caches** as they would require additional cycles, additional pins on the processor chip, and/or additional...

...with the processor chip.

Consequently, it is an object of the invention to provide a **cache** design exhibiting greater hit rates and reduced access times to increase system performance in a...

...addresses this object by providing a data processing system and method thereof utilizing a unique **cache** architecture that performs class prediction in a multi-way set associative **cache** with little or no performance penalty. Consistent with one aspect of the invention, the **cache** architecture may be implemented in a multi-**cache** system, with class prediction for a second **cache** performed at least partially in parallel with handling of a memory access request by a first **cache**. Consistent with another aspect of the invention, the **cache** architecture may be implemented in a **cache** that utilizes a different addressing format from that of a memory access request source coupled...structure. Through this architecture, multiple memory access request sources can access class predict information in **parallel** with reduced latency and low susceptibility to conflicts.

Consistent with yet another aspect of the...

17/3,K/43 (Item 10 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00480749 **Image available**

METHOD, SOFTWARE AND APPARATUS FOR SAVING, USING AND RECOVERING DATA
PROCEDE, LOGICIEL ET DISPOSITIF PERMETTANT DE SAUVEGARDER, UTILISER ET
RECUPERER DES DONNEES

Patent Applicant/Assignee:
WILD FILE INC,

SCHNEIDER Eric D,
FERRIL William C,
WHEELER Douglas N,
SCHWARTZ Lawrence E,
BRUGGEMAN Edward W,

Inventor(s):

SCHNEIDER Eric D,
FERRIL William C,
WHEELER Douglas N,
SCHWARTZ Lawrence E,
BRUGGEMAN Edward W,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9912101 A2 19990311
Application: WO 98US18863 19980904 (PCT/WO US9818863)
Priority Application: US 97924198 19970905; US 9839650 19980316; US
98105733 19980626

Designated States: AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES
FI GB GE GH GM HR HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD
MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG US
US US UZ VN YU ZW GH GM KE LS MW SD SZ UG ZW AM AZ BY KG KZ MD RU TJ TM
AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM
GA GN GW ML MR NE SN TD TG

Publication Language: English

Fulltext Word Count: 51490

Fulltext Availability:

Detailed Description

Detailed Description

... Figure 8 shows how two maps could be used to represent the main area
and **history** buffer.

The mapping scheme allows this method to operate continuously and
maintain old states of...employs the maps to allow for heavy write access
to the disk, but at the **same time**, knowledge of where the main and
extra pages areas are retained. Thus, in the background...

...in performance due to fragmentation from re-mapping.

It is assumed the mapping system is **cached** and efficient so that it
introduces 0 little overhead. Since data is likely written in...

...enough that it is unlikely to fit entirely in RAM.

A root is required, plus **one** mid- **level** node, and 200 low-level nodes
(200,000 entries stored I 000 per node). However...

...of pages allocated in sequential locations, and so the engine is not
constantly hopping from **one** low- **level** node to another.

The upper portion of the tree indicates whether a low-level node...

...In the other 89% of accesses the upper

43

two levels of the tree are **cached** and iminediately indicate
direct, (umnapped) access, adding negligible overhead.

Next consider the context where heavy...the size of the prior, or 28.8
megabytes. This would require one root, plus **two** mid- **level** nodes, and
1,800 low- **level** nodes (1 .8 million entries stored I 000 per node).
Again the top two levels of the tree are generally **cached** . However, now
all accesses go through a low-level node. If in a reasonably worst...

...from the other area. Figure 13 shows the effect of the swapping so that
the **history** map only references pages in the extra page area and the
main map only references...

DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00424596 **Image available**
SELF-TIMED PULSE CONTROL CIRCUIT
CIRCUIT DE COMMANDE D'IMPULSIONS AUTO-CADENCEES

Patent Applicant/Assignee:
ADVANCED MICRO DEVICES INC,

Inventor(s):
PARTOVI Hamid,
HOLST John Christian,
BEN-MEIR Amos,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9815058 A1 19980409
Application: WO 97US17558 19970929 (PCT/WO US9717558)
Priority Application: US 9627329 19960930; US 97938589 19970926

Designated States: AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES
FI GB GE GH HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK
MN MW MX NO NZ PL PT RO RU⁷ SD SE SG SI SK SL TJ TM TR TT UA UG UZ VN YU
ZW GH KE LS MW SD SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH DE DK ES
FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN ML MR NE SN TD
TG

Publication Language: English

Fulltext Word Count: 9372

Fulltext Availability:
Detailed Description

Detailed Description

... unit 1014, issues up to six RISC86 Operations using out-of-order
issuing to seven **parallel** execution units. The execution units
speculatively execute the RISC86 Ops to generate results. The RISC86...
resolving unit 1028. A branch logic unit 1030 implements a branch
prediction operation that uses **two - level** branch prediction based on
an 8192-entry Branch **History** Table (BHT) 1032, a 16-entry Branch Target
Cache (BTC) 1034, and a 16-entry Return Address Stack (RAS) 1036

The dual instruction decoders...engine. The fetch logic 1008 fetches up
to sixteen instruction bytes each cycle from the **L1** instruction **cache**
1002 and transfers the instruction bytes into an instruction buffer (not
shown) preceding the dual...

...instruction bytes from the instruction buffer, decodes up to two X86
instructions, immediately recognizes and **predicts** branches, and
generates up to four RISC86 Ops. The RISC86 Ops are loaded into the...

17/3,K/45 (Item 12 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00424494 **Image available**
MEMORY ARRAY, MEMORY CELL, AND SENSE AMPLIFIER TEST AND CHARACTERIZATION
ESSAI ET CARACTERISATION D'UNE MATRICE MEMOIRE, D'UNE CELLULE DE MEMOIRE ET
D'UN AMPLIFICATEUR DE LECTURE

Patent Applicant/Assignee:
ADVANCED MICRO DEVICES INC,

Inventor(s):
WENDELL Dennis L,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9814956 A1 19980409
Application: WO 97US17635 19970929 (PCT/WO US9717635)
Priority Application: US 9627329 19960930; US 97938257 19970926

Designated States: AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES
FI GB GE GH HU IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN
MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG UZ VN YU ZW
GH KE LS MW SD SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH DE DK ES FI
FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN ML MR NE SN TD TG

Publication Language: English

Fulltext Word Count: 18928

Fulltext Availability:

Detailed Description

Detailed Description

... obtained from a processor system bus interface 1930 are pre-decoded during filling of a **level** -one (LI) instruction cache 1902 after which the predecode bits are stored in a predecode...blocks. The result is a 256 x 2 x 64, or 32 KB cache.

A **level** - **one** cache controller 1940 controls caching in instruction **cache** 1902. Instruction **cache** 1902 uses a most recently used scheme (MRU) to predict the way selection on **cache** accesses. A misprediction in the way selection causes a one cycle penalty. Instruction **cache** 1902 uses a least recently used (LRU) line replacement algorithm. An alternative random replacement algorithm is supported through a configuration bit. Instruction **cache** 1902 also supports a direct-mapped replacement algorithm, although using the configuration reduces the **cache** size from 32 KB to 16 KB. Instruction **cache** 1902 performs a simple prefetching algorithm. When a line miss occurs, as distinguished from a...

...is 0), then both sub-blocks are fetched and pipelined on the bus. The data **cache** 1922 includes a 128-entry data translation lookahead buffer (DTLB). In contrast to instruction **cache** 1902, the data **cache** 1922 uses a least recently missed (LRM) selection technique which is generally a more accurate...

...scheme than the LRU technique. In the LRM scheme, the line that first enters the **cache** is replaced. An alternative random replacement algorithm is also supported. The data **cache** 1922 also supports a direct-mapped replacement algorithm, reducing the cache size from 32 KB ...

...retires the results in order. Branch unit 1920 implements a branch prediction operation that uses **two** -level branch prediction based on an 8192-entry branch history table (BHT), a 16-entry...

...instruction bytes from the instruction buffer, decode up to two x86 instructions, immediately recognize and **predict** branches, and generate up to four RISC86 Ops. The RISC86 Ops are loaded into the...

17/3,K/46 (Item 13 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

(c) 2004 WIPO/Univentio. All rts. reserv.

00424493 **Image available**

DATA RETENTION TEST FOR STATIC MEMORY CELL

ESSAI DE RETENTION DE DONNEES POUR CELLULE DE MEMOIRE STATIQUE

Patent Applicant/Assignee:

ADVANCED MICRO DEVICES INC,

Inventor(s):

WENDELL Dennis L,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9814955 A1 19980409

Application: WO 97US17552 19970929 (PCT/WO US9717552)

Priority Application: US 9627329 19960930; US 97938732 19970926

Designated States: AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES

FI GB GE GH HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK

MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG UZ VN YU

ZW GH KE LS MW SD SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH DE DK ES

FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN ML MR NE SN TD

TG

Publication Language: English

Fulltext Word Count: 18794

Fulltext Availability:
Detailed Description

Detailed Description

... obtained from a processor system bus interface 1930 are pre-decoded during filling of a **level -one** (LI) instruction cache 1902 after ... blocks. The result is a 256 x 2 x 64, or 32 KB cache.

A **level - one** cache controller 1940 controls caching in instruction **cache** 1902. Instruction **cache** 1902 uses a most recently used scheme (MRU) to predict the way selection on **cache** accesses. A misprediction in the way selection causes a one cycle penalty. Instruction **cache** 1902 uses a least recently used (LRU) line replacement algorithm. An alternative random replacement algorithm is supported through a configuration bit. Instruction **cache** 1902 also supports a direct-mapped replacement algorithm, although using the configuration reduces the **cache** size from 32 KB to 16 KI3. Instruction **cache** 1902 performs a simple prefetching algorithm. When a line miss occurs, as distinguished from a...is 0), then both sub-blocks are fetched and pipelined on the bus.

The data **cache** 1922 includes a 128-entry data translation lookahead buffer (DTLB). In contrast to instruction **cache** 1902, the data **cache** 1922 uses a least recently missed (LRM) selection technique which is generally a more accurate...

...scheme than the LRU technique. In the LRM scheme, the line that first enters the **cache** is replaced. An alternative random replacement algorithm is also supported. The data **cache** 1922 also supports a direct-mapped replacement algorithm, reducing the cache size from 32 KB ...

...retires the results in order. Branch unit 1920 implements a branch prediction operation that uses **two** -level branch prediction based on an 8192-entry branch history table (BHT), a 16-entry...

...instruction bytes from the instruction buffer, decode up to two x86 instructions, immediately recognize and **predict** branches, and generate up to four RISC86 Ops. The RISC86 Ops are loaded into the...

17/3,K/47 (Item 14 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00196049 **Image available**

TWO-LEVEL BRANCH PREDICTION CACHE

ANTEMEMOIRE DE PREDICTION DE BRANCHEMENT A DEUX NIVEAUX

Patent Applicant/Assignee:

NEXGEN MICROSYSTEMS,

Inventor(s):

STILES David R,

FAVOR John G,

VAN DYKE Korbin S,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9113402 A1 19910905

Application: WO 91US826 19910206 (PCT/WO US9100826)

Priority Application: US 90306 19900226

Designated States: AT BE CH DE DE DK ES FR GB GB GR IT JP KR LU NL SE

Publication Language: English

Fulltext Word Count: 9090

Fulltext Availability:
Detailed Description

Detailed Description

... be made, or normal processing, with the attendant possibility of incurring delays, must be performed.

Past high performance CISC designs have used forms of cache structures to hold various combinations of...information. This is offset by the much larger number of entries, compared to the first **level** cache, which are capable of implementation with a given amount of hardware circuitry.

The size...

...especially makes sense given that the second level cache serves as a backup to the **first level** cache. With this much larger size, even given the direct-mapped organization, the second level cache provides an effective backup to the **first level** cache.

In the preferred embodiment of this invention, a second level cache entry holds only...address bits from the branch instruction's address with the lower 16 bits from the **cache** entry.

The second level BPC uses a direct-mapped access method, versus a set or fully associative method. This is acceptable due to the relatively large number of second level **cache** entries. Even more significantly, it is then possible to discard the tag and tag storage associated with each **cache** entry. In essence, when a **cache** look-up accesses a selected entry, it is simply assumed that the tag and look...

...incorporating the present invention; Fig. 2 is an overall block diagram of the branch prediction **cache** (BPC) and its immediate environment;

Fig. 3 is a block diagram of the **first level** BPC

Figs. 4-8 are logic schematics of the various memory arrays and logic within the **first level**

BPC; and

Fig. 9 is a logic schematic of the second level BPC.

DESCRIPTION OF...

...of up to three simultaneous instruction streams. DEC 12 contains a two-level Branch Prediction **Cache** (BPC) 13. The BPC includes an integrated structure which contains dynamic branch history data, a...

...are decoded, the BPC is consulted for information about that branch. Independent of the direction **predicted**, branches are executed in a single cycle and do not cause pipeline bubbles. On each...according to the present invention. In accordance with the present invention, the BPC is a **two**-level structure including a first level BPC 152 and a second level BPC 155. As...is conditional, the branch target address, and cached target instruction data. More specifically, each first **level** cache line contains the target address from when the branch instruction was last executed; up...

...recording the direction taken during the past executions of the branch instruction.

To this end, **first level** BPC 152 includes a program counter cache (PCC) preferably implemented as a program counter content...

...select inputs (as opposed to encoded word address inputs). Some of these sets of A **cache** tag for each entry, namely the instruction address of the branch associated with the entry, is stored in PcCAM 170. A **first level** **cache** look-up is performed by accessing PcCAM 170 using the above next instruction address, and...

...for which there was a tag match.

This greater associativity is with respect to both **cache** look-ups and **cache** replacements, i.e. when adding each entry to the **cache** a new entry to the **cache** requires that some other (hopefully less beneficial) entry be removed to make room. Through the...